85/90

Image classification on CIFAR dataset

Abstract

Image classification is one of the core topics in Computer Vision. Image Classification is nothing but assigning an image to one of the labels from a fixed set of categories. When we perform image classification, our application will receive an image as input and the system will be aware of the set of categories and tries to assign a category label to that particular image.

The task is simple in terms of human perspective, but in fact difficult when it comes to computers because computers don't know the image it reads the images in form of grids a 3d array of integers ranging from 0 - 255. the 3 dimensions are height, width, and channels. Mostly color images have 3 channels i.e., red, blue, and green. Black and white images have 1 channel either 0, or 1.

Due to improvements in technology. Convolution Neural networks are known for image recognition, so they are better used for such image classification purposes. Typically, a CNN has Convolution layers, Relu Layers, pooling layers, and fully connected dense layers. Image Recognition is the root of many improvements in computer vision problems like movement prediction, video recognition, facial or pattern recognition, etc.

Introduction

As Image recognition is the new revolutionary sector in computer vision, the study in image recognition is also increasing day by day as there exist many advantages of image recognition like providing safety features in cars that detect large objects, that can help the visually impaired or the detection of water on planets in outer space. It can be anything. We are much bounded to have such technology that has numerous effects on our daily life. Image classification is one step toward the recognition of the object, when one particular set of objects is classified into different categories we can then use them for later uses. For example, if any computer or machine identifies the animal horse/cat then it can perform certain

Report: 30
Understanding/implementation: 55

By Akhil Raj Tirumalasetty (at21bo) & Sarthak Sharma (ss20nc)

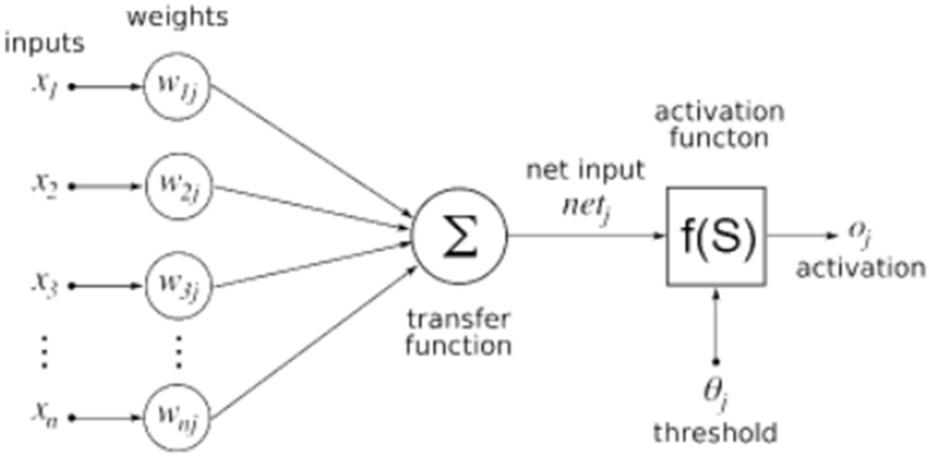
I would expect some uncommon components.

tasks that humans need not perform like feeding them, such scenarios ease the human effort.

There were many models till now that work tremendously on classifying the object. Among them, the first ImageNet was launched by the scientists of Princeton and Stanford in the year 2009, with close to 80,000 keyword-tagged images, which has now grown to over 14 million tagged images. All these images are easily accessible at any given point in time for machine training. On the other hand, Pascal VOC is powered by numerous universities in the UK and offers fewer images, however, each of these come with richer annotation. This rich annotation not only improves the accuracy of machine training but also paces up the overall processes for some applications, by omitting a few of the cumbersome computer subtasks.

In our project, we propose to build a simple deep learning model that can identify a particular set of images and categories them. So that in the future whenever they see any real-time such object/living being they accurately identify them. Deep learning is the concept of machine learning but has a neural architecture quite similar to the human brain.

Neural networks can be considered as weighted directed graphs where neurons act like nodes and connections between two nodes are the weighted edges. The processing of a neural network is the processing of a neuron that receives a signal from many inputs (mostly from the outer world).

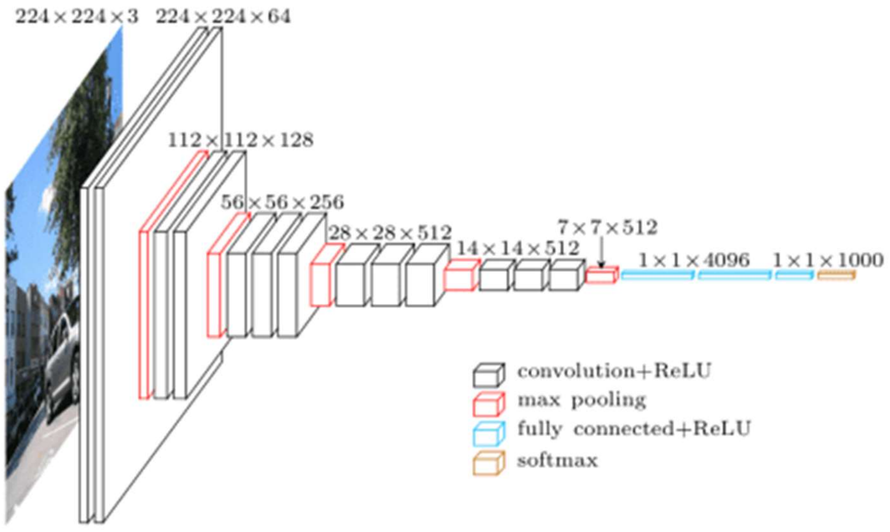


Simple Neural Network

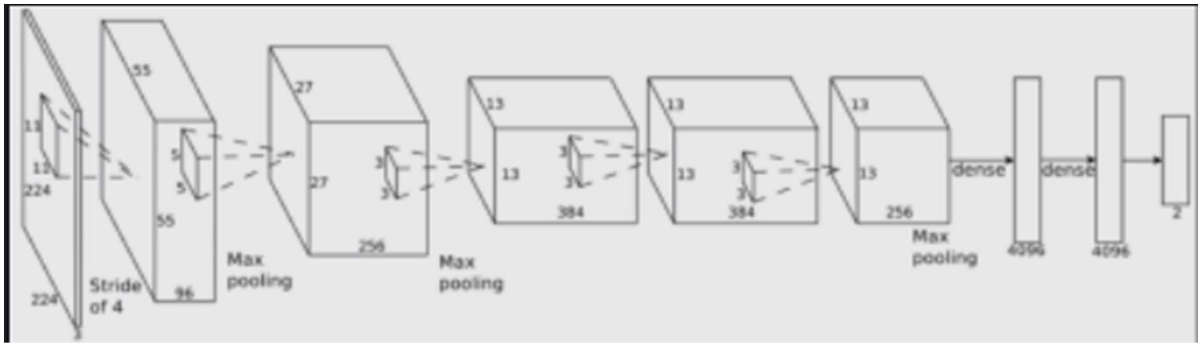
Here we try to give our image as input to the neural network and perform certain actions on the data internally inside the hidden layers and then give an output that classifies our image into one of the categories we provided. We run this process multiple times so that the model adapts the process and adjust its weights such that it gives correct results next time. Such process of input from start of node till getting the output from last node is called one epoch and the amount of information we send to the nodes is often referred to the batch size.

We try to build a good convolution neural network that performs well by supervised learning. Supervised learning is nothing but providing the input images and their output as well, this will let the model to memories certain images and understand them. For this, we are using the CIFAR dataset. Classification problems use an algorithm to accurately assign test data into specific categories, such as separating apples from oranges. Or, in the real world, supervised learning algorithms can be used to classify spam in a separate folder from your inbox. Linear classifiers, support vector machines, decision trees, and random forest are all common types of classification algorithms. For unsupervised learning, we'll not provide any input to the model while the model itself tries to figure out the solution for the problem.

Typical architectures of Imagenet



VGG network architecture

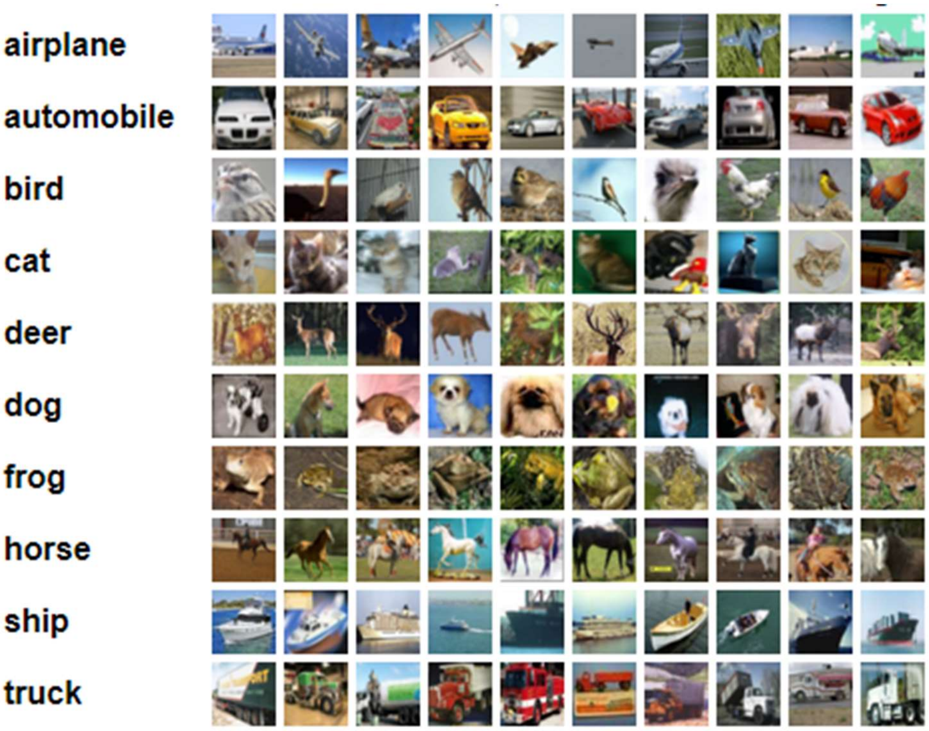


Alexnet

Dataset (CIFAR)

We are using a dataset called CIFAR which has 60,000 images classified into 10 categories most commonly known as CIFAR-10. These images are 32x32 color images divided into 50000 images as training images and the remaining 10000 images for testing.

These classes are completely mutually exclusive. No image will contain 2 or more different objects in it. They have the following class labels



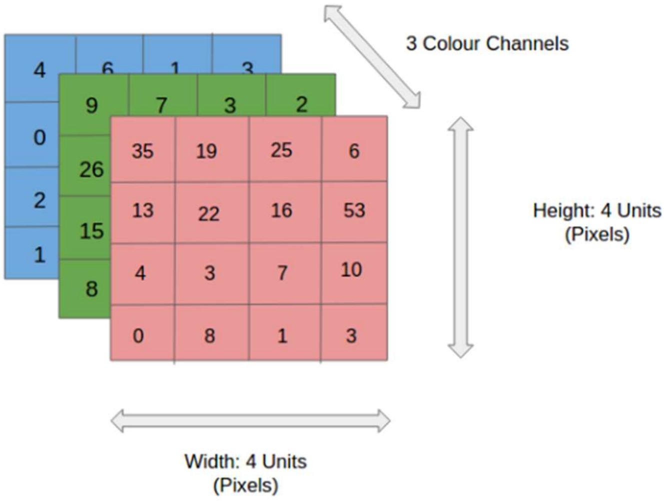
CIFAR 10 Class Labels

CIFAR - 100 is also similar to CIFAR-10 but in an updated version of it. It has 100 class labels instead of 10. These are the following class labels for it.

beaver, dolphin, otter, seal, whale, fish, flatfish, ray, shark, trout, orchids, poppies, roses, sunflowers, tulips, bottles, bowls, cans, cups, plates, apples, mushrooms, oranges, pears, sweet peppers, clock, computer keyboard, lamp, telephone, television, bed, chair, couch, table, wardrobe, bee, beetle, butterfly, caterpillar, cockroach, bear, leopard, lion, tiger, wolf, bridge, castle, house, road, skyscraper, cloud, forest, mountain, plain, sea, camel, cattle, chimpanzee, elephant, kangaroo, fox, porcupine, possum, raccoon, skunk, crab, lobster, snail, spider, worm, baby, boy, girl, man, woman, crocodile, dinosaur, lizard, snake, turtle, hamster, mouse, rabbit, shrew, squirrel, maple, oak, palm, pine, willow, bicycle, bus, motorcycle, pickup truck, train, lawn-mower, rocket, streetcar, tank, tractor

Convolution Neural Networks

CNNs are a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and differentiate one from the other. When compared to other classification algorithms, the amount of pre-processing required by a ConvNet is much lower. While filters in primitive methods are hand-engineered, ConvNets have the ability to learn these filters/characteristics with enough training. A ConvNet's architecture is similar to the connectivity pattern of neurons in the human brain and was inspired by the organization of the Visual Cortex. Individual neurons only respond to stimuli in a narrow region of the visual field known as the Receptive Field. A group of such fields' overlaps to cover the entire visual area. Some layers used in our model.



4×4×3 RGB Image

Convolution layers:

After passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation map, with shape:

(number of inputs) x (feature map height) x (feature map width) x (feature map channels).

Convolutional layers convolve the input and pass the result to the following layer. This is analogous to a neuron's response to a specific stimulus in the visual cortex. Each convolutional neuron only processes information for its own receptive field. Although fully connected feedforward neural networks can be used to learn features and classify data, they are generally impractical for larger inputs such as high-resolution images. Due to the large input size of images, where each pixel is a relevant input feature, it would require a very high number of neurons, even in a shallow architecture.

Pooling layers:

Along with traditional convolutional layers, convolutional networks may include local and/or global pooling layers. By combining the outputs of neuron clusters at one layer into a single neuron in the next layer, pooling layers reduce the dimensions of data. Pooling, which is a type of non-linear down-sampling, is another important concept in CNNs. Pooling can be implemented using a variety of non-linear functions, the most common of which is max pooling. It divides the input image into rectangles and outputs the maximum for each of these sub-regions.

Intuitively, a feature's exact location is less important than its approximate location in relation to other features. This is the main reason for using pooling in convolutional neural networks.

Hyper parameters are various settings that are used to control the learning process. Some Hyper parameters are given as follows.

Kernel

The kernel is the number of pixels processed together

Padding

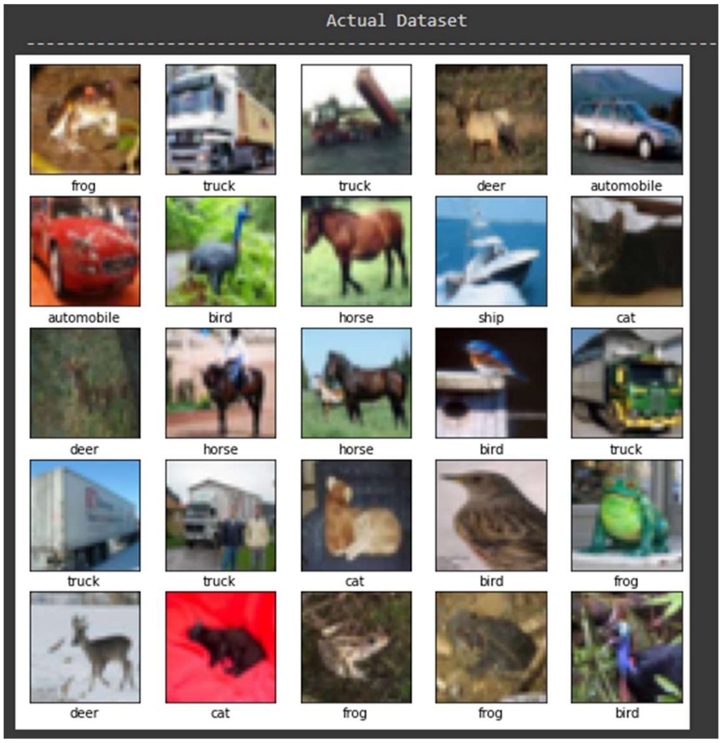
Padding is the addition of (typically) 0-valued pixels to an image's borders. This is done to prevent the border pixels from being undervalued (lost) in the output because they would normally only participate in a single receptive field instance. Padding is usually one less than the corresponding kernel dimension. A convolutional layer with 3x3 kernels, for example, would receive a 2-pixel pad on all sides of the image.

Stride

The stride is the number of pixels moved by the analysis window per iteration. A stride of 2 indicates that each kernel is 2 pixels offset from its predecessor.

Implementation

As mentioned, the images are loaded using CIFAR dataset. As the images are nothing but the 3d array of the values ranging 0-255, normalized the data to be in range 0-1 and class labels (Y) are one hot encoded.



Data after loading the dataset.

Architecture

The Model is built using the following architecture

[32C3-32C3-P2] - [64C3 – 64C3-P2] - [128C3 – 128C3 – P2] along with Dropout and Batch Normalization. L2 regularizer with weight decay value of 1e-3.

Layer	Output Shape
(Conv2D)	(None, 32, 32, 32)
Activation	(None, 32, 32, 32)
Batch Normalization	(None, 32, 32, 32)
(Conv2D)	(None, 32, 32, 32)
Activation	(None, 32, 32, 32)
Batch Normalization	(None, 32, 32, 32)
Max Pooling	(None, 16, 16, 32)
Drop out	(None, 16, 16, 32)
(Conv2D)	(None, 16, 16, 64)
Activation	(None, 16, 16, 64)
Batch Normalization	(None, 16, 16, 64)
(Conv2D)	(None, 16, 16, 64)
Activation	(None, 16, 16, 64)
Batch Normalization	(None, 16, 16, 64)
Max Pooling	(None, 8, 8, 64)
Drop out	(None, 8, 8, 64)
(Conv2D)	(None, 8, 8, 128)
Activation	(None, 8, 8, 128)
Batch Normalization	(None, 8, 8, 128)
(Conv2D)	(None, 8, 8, 128)
Activation	(None, 8, 8, 128)
Batch Normalization	(None, 8, 8, 128)
Max Pooling	(None, 4, 4, 128)
Drop out	(None, 4, 4, 128)
Flatten	(None, 2048)
Dense	(None, 10)

Total params	309,290
Trainable params	308,394
Non-trainable params	896

Along with a good architecture, LR decay is also implemented to obtain best performance. Activation function used in this model is ELU.

In order to maintain a good generalization, Data Augmentation is used ImageDataGenerator is used to alter the original images into its rectified versions.

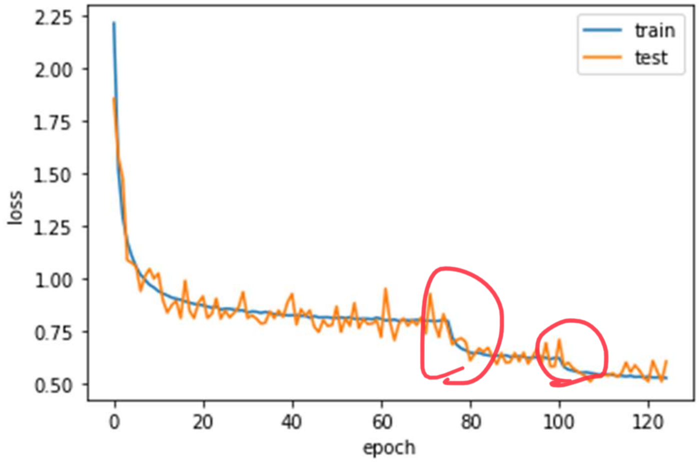
Finally, the Model is compiled along with RMS optimizer starting with a learning rate of 0.001

Results

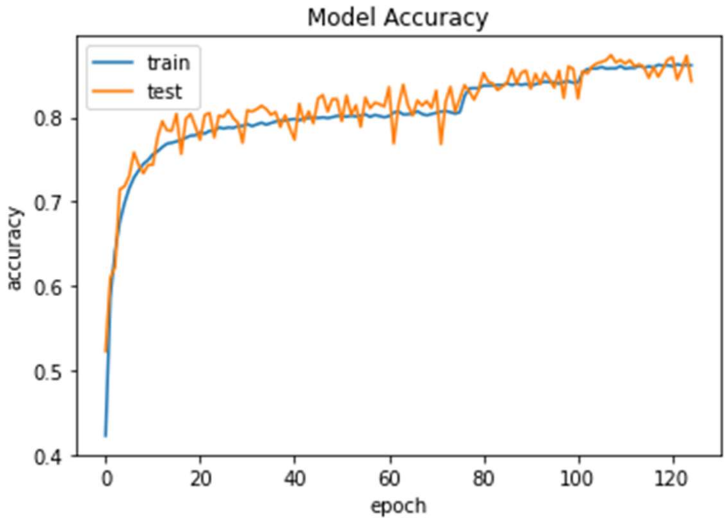
After running the model with batch size of 64 and 125 epochs.

For CIFAR-10. Achieved a final accuracy percentage of 87.3 %

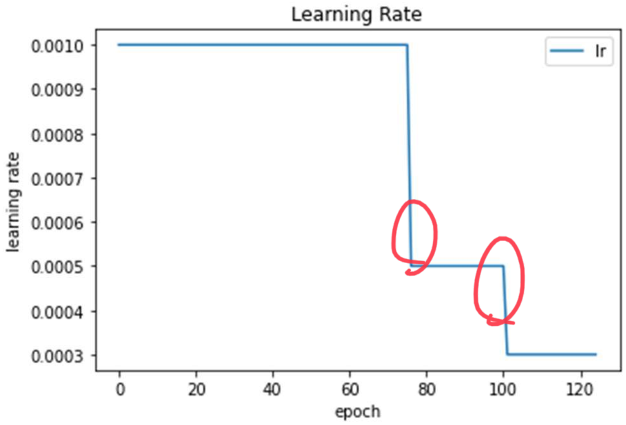
Loss throughout the model



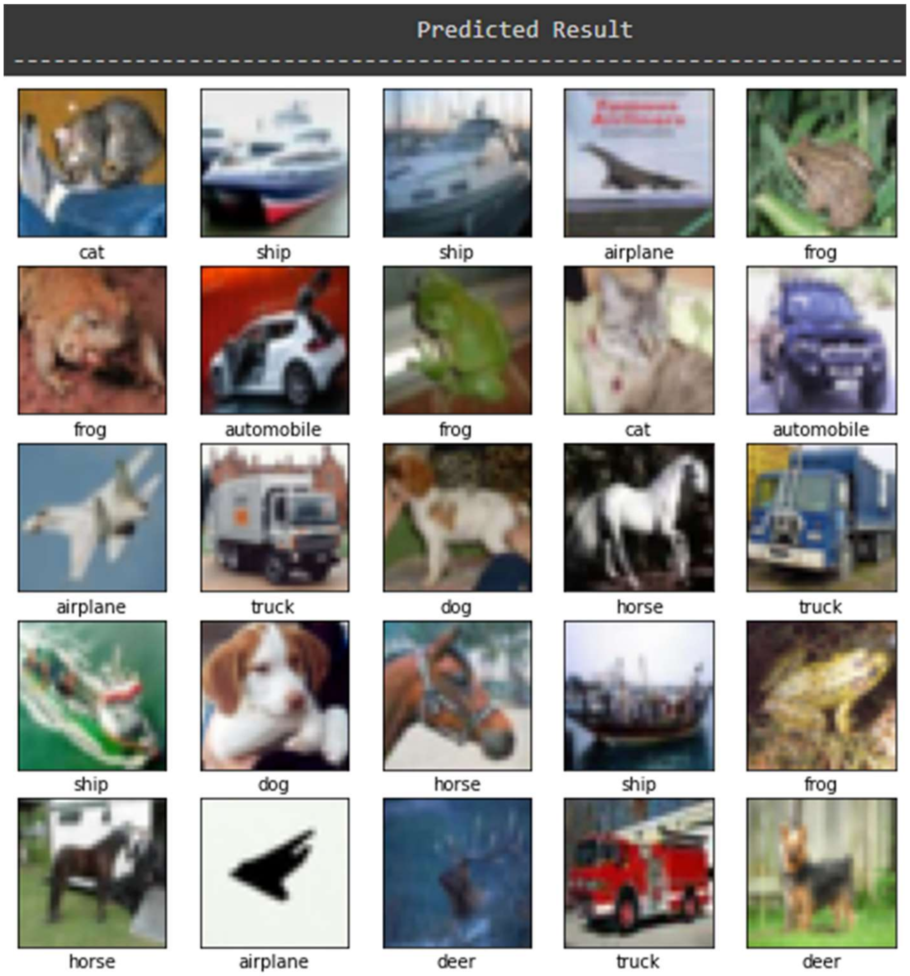
Accuracy of the model



Learning Rate



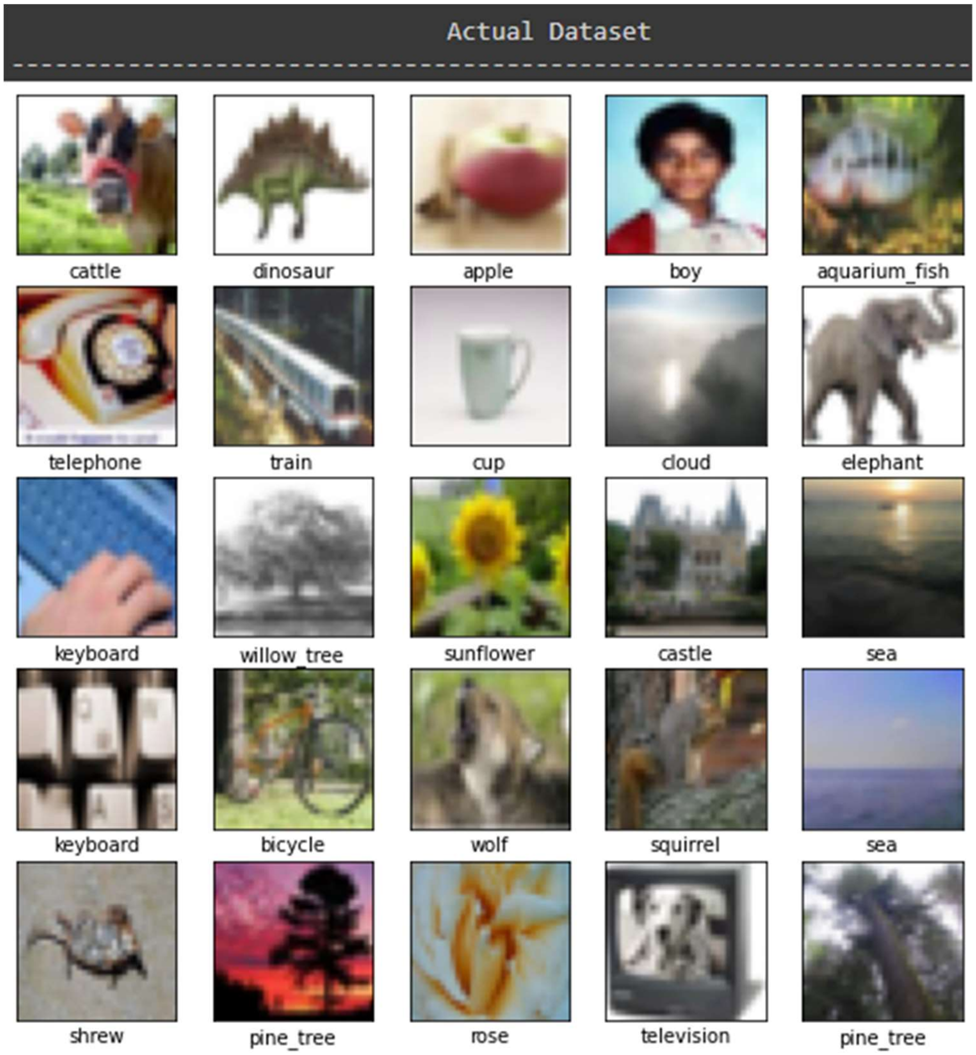
Images are classified as follows:



CIFAR 10, Classification Results

For CIFAR 100,

Data loaded before running the model fit.

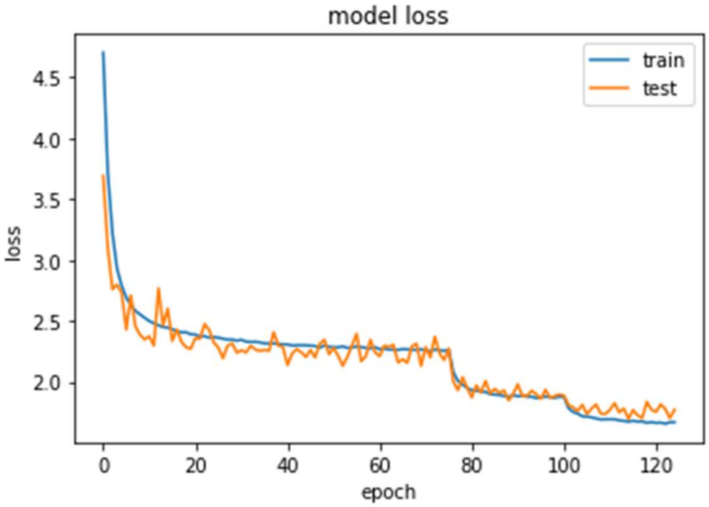


Final Accuracy percentage achieved is 71%, As there are 100 class labels it requires more number of epochs and training time. As we can see in the images they have the size of 32x32x3.

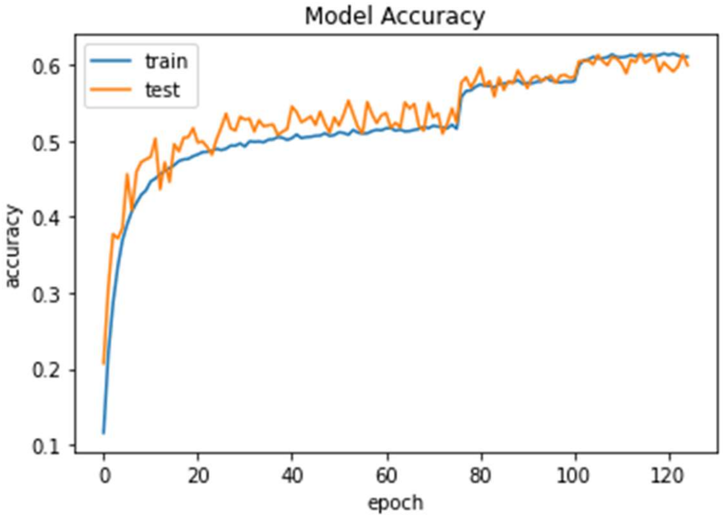
The model also follows L2 regularizer with weight decay rate of 1e-3. Augmented data is passed to the model before fit. L2 regularization along with data augmentation helped us to overcome the over-fit of data.

The initial learning rate provided to the RMSprop is 0.001 with decay of 1e-6 and compiled with categorical cross-entropy as we are classifying into 10/100 labels.

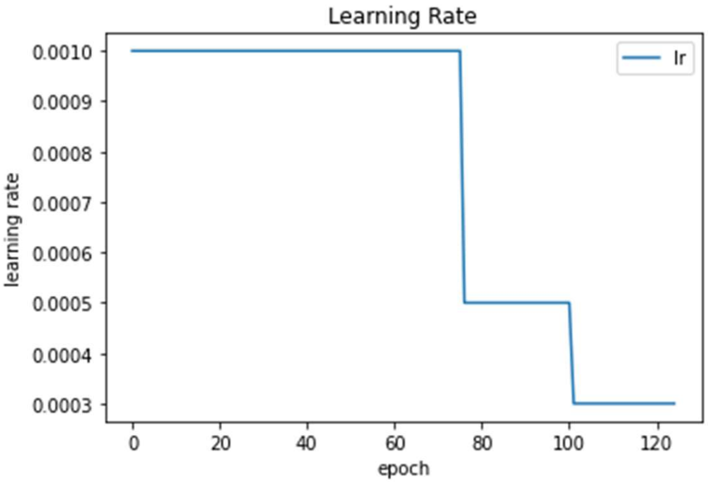
Loss of the Model



Accuracy of the Model

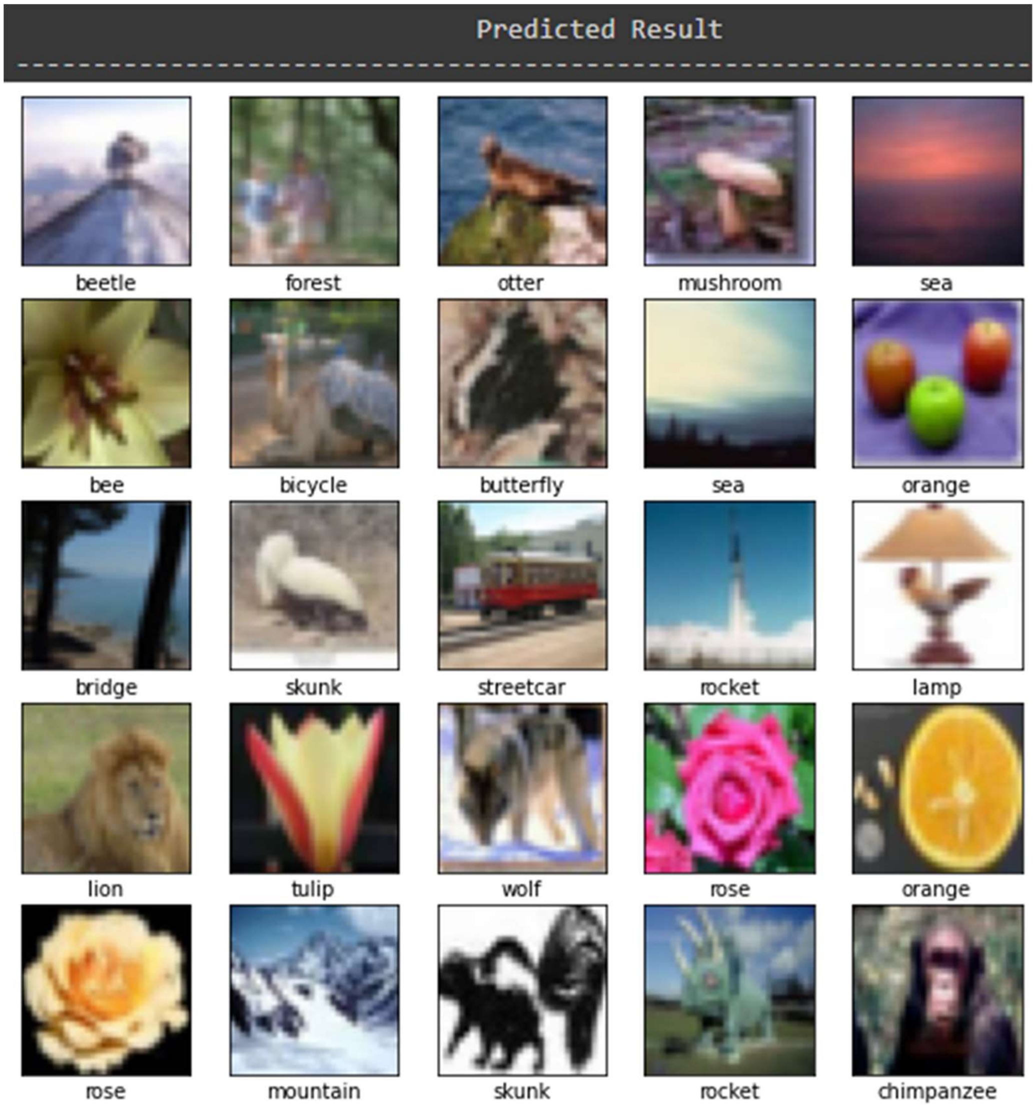


Learning Rate (step decay)



By Akhil Raj Tirumalasetty (at21bo) & Sarthak Sharma (ss20nc)

Images are classified as follows:



```
                              Predicted Result
--------------------------------------------------------------------
```

Classified Results for CIFAR 100

Conclusion

Overall the performance of the model was good, the architecture of the model has maintained at least not worst but a good state-of-the-art level of accuracy and loss values. This accuracy of the model is achieved by tuning hyper parameters multiple times and we also found that normal images directly passed on to the model give less accuracy when compared with rectified images with alterations like flip, tilt, etc. Drop out and Batch Normalization techniques gave us a way to overcome the overfitting. If you see the results on the validation set, they do not suffer from any overfitting.

This model can perform more in the future if we implement a better architecture than now and tune hyper parameters more, the images could now be loaded in real-time and the model can directly classify such objects, but limited to few class labels. If the dataset grows more in the aspect of number and classes, then we can achieve a model that has revolutionary advantages in the field of computer vision.

References

[1] Russakovsky, O., Deng, J., Su, H. et al. ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis 115, 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y

[2] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge 2010. www.imagenet.org/challenges. 2010.

[3] D.C. Cire¸san, U. Meier, J. Masci, L.M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. Arxiv preprint arXiv:1102.0183, 2011.

[4] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.

[5] D. Cire¸san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745, 2012.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

[7] Zeiler, M.D., Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53

[8] Hinton, Geoffrey (2012). "ImageNet Classification with Deep Convolutional Neural Networks". NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. 1: 1097–1105 – via ACM